AN APROACH TO NATURAL-LANGUAGE SEMANTICS IN LOGIC PROGRAMMING

PATRICK SAINT-DIZIER

▷ The author presents the main lines of an approach to the semantics of natural-language sentences within the logic programming framework. A formalism and tools are defined to represent noun phrases, verb phrases, adjectives, determiners, and some adverbs with a degree of precision deemed suitable for a natural-language front end. The formalism used is based on first-order logic augmented by PROLOG calls. A methodology is presented rather than an exhaustive list of results. The author also gives the main lines of an approach to building the semantic representation of sentences, i.e., by the application of rewrite rules associated with contextual preconditions of application. ◁

1. INTRODUCTION

The long-term objective of this work is the specification of a friendly man-machine interface that supports natural-language communication. Such an interface has to be robust, helpful to the user, and transportable [33]. It has also to take into account a large variety of linguistic phenomena and to deal with them in a way and with a degree of precision that we think adequate and relevant for a natural-language front end.

In this work, we use formalisms and tools developed within the logic-programming framework as a conceptual basis. We do not dwell here on the interest of using logic programming for natural-language processing, since other recent works are sufficiently eloquent about this point [7, 25, 14, 31, 30]. Logic can be used as the underlying formalism of the parser and of the component that computes the

Address correspondence to Patrick Saint-Dizier, I.R.I.S.A., Campus de Beaulieu, 35042 Rennes-Cedex, France.

Received 1 April 1985; accepted 1 May 1986.

THE JOURNAL OF LOGIC PROGRAMMING

52 Vanderbilt Ave., New York, NY 10017 0743-1066/86/$03.50

semantic representation of a sentence as well as the formalism of the semantic representation itself and of the programming language. This results in a greater uniformity and simplicity of expression.

Natural-language analysis is used here for a practical purpose, namely interacting with a knowledge base via requests expressed in a limited domain of a language. We view a knowledge base as a finite set of statements about a precise domain, expressed by facts, rules, and integrity constraints. We think that the use of a limited domain is not a real restriction, since humans, at a given moment and about an objective domain, process and store limited resources. Within our framework, a knowledge base has the general form of a deductive database [15, 22].

This point of view taken here is slightly different of that of Montague [11], whose model-theoretic approach leads to very complex computational problems. This does not mean that I think that Montague's semantics is too formal, complicated, and constraining for natural-language processing. I think that Montague's semantics reflects, in fact, the complexity of natural-language processing and understanding, but the means of expression which are adopted cannot, in my opinion, be used directly for our purpose. This remark is also valid for the discourse-representation structures of Kamp [20]. On the other hand, our approach is closer in its spirit to recent computational developments in situation semantics [28].

In this paper, I present the main lines of an approach to the semantics of natural-language sentences in logic programming. Surprisingly enough, much less attention seems to have been devoted to the semantic representation of sentences than to their syntactic analysis. To my knowledge, the only exceptions are the work of Colmerauer [4], Dahl [5–7], Pereira [31], McCord [24–26], and Filguieras and Porto [12]. They propose elegant and powerful formalisms to deal with several crucial points about semantics, which we will refer to in the next sections. Most of the other work, as far as I know, uses an ad hoc representation, highly dependent on the domain.

Our starting point is the formalism of the three-branched quantified tree developed by Dahl [5] and suggested to a certain extent by Colmerauer [4]. Our goal is to go deeper in the semantic representation of a sentence than the previous work does, where some aspects of the representation remain superficial and thus inefficient and useless in real man-machine communication. However, it should be noticed that we introduce in this work a methodology rather than an exhaustive formalism. This work is a contribution to the study of the representation of entities such as determiners, adjectives, and adverbs, and of some structures such as negation and complex noun phrases. We develop a quite small set of tools, based on set theory, which we consider to have an adequate degree of precision for our purpose. These tools are used by human experts when they define the sematics of each lexical entity that occurs in the specific domain they consider. They are easily transportable to various domains of discourse and also to various formalisms such as those developed by Pereira [31], McCord [25], Palmer [30], and Filguieras and Porto [12] within the logic-programming framework.

All the tools and the formalisms we present here are directly interpretable by PROLOG. In particular, one of our main goals is to reduce higher-order predicates into first-order predicates augmented by PROLOG calls or to use well-defined extensions to first-order logic for which suitable proof procedures are available. This paper presents the target semantic representation of our system. Beyond the scope

of the present paper is, in particular, the specification of interactions with the context when several representations of a sentence are possible. Notice that the more precise the semantic representation is, the more complex and context-dependent is its computation. There is, in fact, a compromise to be made between the degree of precision of the semantic representation of a sentence and the degree of complexity involved in its computation.

The next section presents the general framework of our approach and introduces the general tools we need in the remainder of this paper. Section 3 gives the basic semantic representation of nouns, noun complements, verbs, and negation. Next, Section 4 is devoted to the semantics of determiners, and Section 5 to the semantics of adjectives. Finally, Section 6 introduces some ideas on the way to represent some of the most common adverbs.

2. CONTEXT AND TOOLS

In this section, we first consider how we view the process of natural-language understanding. Next, the basic formalism of the semantic representation is introduced. Finally, we define the general tools we use in the next sections to describe the semantics of lexical items.

2.1. General Approach to Natural-Language Understanding

We view the natural-language understanding process as the application of a complex function H that realizes the transformation of an external form into a certain understanding in a given domain of knowledge. A strategy to define H is to decompose it into a linear sequence of functions h_i that apply on intermediate structures S_i:

$$H(P) = h_n \circ h_{n-1} \circ \cdots \circ h_1(P).$$

The decomposition is motivated by linguistic and mathematical considerations. Then, for computational reasons, the h_i may be again decomposed or, conversely, integrated.

The exact nature of each S_i and h_i is not yet completely clear to us. However, within the logic-programming framework, we view the h_i as rewrite systems. For instance, the syntactic analysis process rewrites a sentence into a syntactic tree. Then (roughly speaking) a semantic interpretation process rewrites the syntactic tree into a logic formula. Finally, this logical formula is rewritten into a set of PROLOG clauses.

A model we think interesting to deepen is to represent each h_i by a set of rewrite rules (possibly of type zero and nondeterministic) with preconditions of application. The preconditions of application basically involve contextual knowledge. By contextual knowledge, we mean knowledge about the specific domain which is considered as well as common-sense reasoning knowledge and general syntactic knowledge (for instance, constraints on movement of constituents [10], or for the reconstruction of ellipsis [32]).

In other terms, all the transformations of an input structure S_j into an output structure S_{j+1} are expressed by rewrite rules, and then preconditions give the domain of application of each rule. This approach seems to be relevant for instance to dealing with quantifier scoping [34]. It appears that the basic elements are the intermediate structures S_i and the functions h_i are only the means to reach S_{i+1} from S_i.

An important point and a major difficulty is to make a distiction between preconditions which are general (i.e. domain-independent and, thus, transportable) and those which are not. It appears to be very difficult to add domain-dependent preconditions to an existing system, and thus much effort has to be devoted in the future to find general domain-independent preconditions and mechanisms to build them. For instance (roughly speaking) when there are several possibilities to represent an NP or a VP, one way to select the correct representation is to select the representation (1) that does not contradict any integrity constraint and (2) whose evaluation succeeds according to rules and facts of the knowledge base.

The main advantages of such an approach are:

modularity at global and local levels,

inheritance of the precise semantics and of the properties of rewrite systems,

direct implementation in PROLOG of rewrite rules associated with preconditions of application.

Two quite different approaches to the use of contextual information to build the semantic representation of a sentence were proposed by M. Palmer in [30] and M. McCord in [26]. In [30] calls to the context are done at the level of lexical entries. An original methodology is introduced that gives the preponderance of domain specific knowledge and semantics over general knowledge. The overall structure of a sentence is built around the main verb of the sentence, which indicates the action(s) to be performed. In [26], M. McCord describes a semantic interpretation component SEM. In SEM, there is an interleaving of several different processes: sense selection for a word, slot filling, movement of nodes (raising and reordering), simplifications, and exercising of semantic constraints that involve real-world knowledge checking. Finally, a formalism to describe the different ways of building the semantic representation of a sentence is described by H. Abramson in [1]. The way to build the representation of a sentence is specified by one or more rules in the form of Horn clauses attached to each node of the parse tree. We think that these approaches are a good experimental and theoretical basis to develop more complex semantic interpreters.

2.2. Characteristics of the Input Representation

The representation of sentences we use as the starting point to our work is the three-branched quantified tree representation introduced and used by Colmerauer [4] and Dahl [5–7], but without taking into account the quantifier heirarchy rules in these works. We adopt the quantifier hierarchy rules described in [34].

Very briefly, in this representation, referential words (nouns, adjectives, and verbs) are translated into predicates whose arguments are typed. Determiners

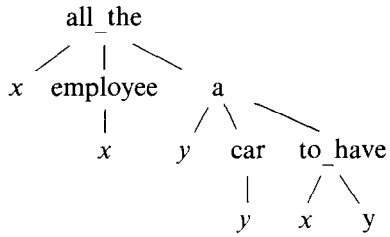
introduce a kind of metapredicate whose arguments are predicates of the form

$$\text{Det}(x, \text{Restr}(x), S) \quad \text{or} \quad \begin{array}{ccc} & \text{Det} & \\ x & \text{Restr} & S \\ & | & \\ & x & \end{array}$$

where Det is a determiner, Restr is the NP's translation (or part of it), and S is the remainder of the sentence which is translated itself in the same manner. The subject NP is first translated and, thus, appears at the top of the tree, then complements in their input reading order and the verb are translated. Notice that Restr can include the representation of adjective phrases, noun complements, and relative clauses. Thus, the sentence

"All the employees have a car."

is represented by



This representation can directly be transformed into the PROLOG subset of first-order logic, and thus it is directly evaluable.

We consider here this representation as the result of the parsing process. We will show in the next sections how each node of the tree is rewritten into a more precise representation that can be a single predicate or a whole subtree.

In our system, lexical entries of nouns, adjectives, and verbs have the form of inference rules, as in [30]. Producing a semantic representation of a particular entity in a sentence is, from a certain point of view, proving that it has been used appropriately for the domain considered. PROLOG is the ideal language for such an approach. The form of lexical entries also owes much to the spirit of lexicon grammars [16] for sense disambiguation and selection.

2.3. General Tools

The tools we present here will be used and illustrated in the next sections. What should be kept in mind is the way these tools are defined, i.e. in the spirit of practical tools, directly executable in PROLOG. The reader is supposed to be familiar with the main PROLOG calls (set_of, card, list_of, etc.).

Definition 1. A property prop is measurable if and only if, given an object x or a set of objects s, it assigns to x (or to s) a value $x1$, and $x1$ belongs to an (ordered) subset of reals or integers. It is written as

$$\text{prop}(x, x1) \quad \text{or} \quad \text{prop}(s, x1).$$

We consider the cardinal of a set [written card(s, x1)] to be a measurable property of this set.

Definition 2. For a measurable property prop possessed by some of the elements x of a set s [i.e., prop(x, x1) is known to the knowledge base], it is possible to compute the average value of prop for all the x. The predicate AVERAGE computes this average value. It is written

AVERAGE(prop, x, def, val)

where

- prop is the name of the property (weight, height, etc.; it is expressed by a predicate known to the knowledge base),
- x is a variable that represents any object of the set s,
- def is the definition of the set of objects we consider [for convenience, this argument can be either a set former such as set_of (x, P, s), where P is a logical formula, or an explicit set of objects],
- val is a real.

Here val is the average value of the $x1$ computed from all the objets of the set s for which prop(x, $x1$) exists. When a formula contains a predicate AVERAGE, then AVERAGE is always true except if s is the empty set. An example of an average value is the height of French men:

AVERAGE(height, x, set_of(x, AND(men(x), french(x)), s), val)

where val is instantiated to the average height of French men.

It should be noted that we also allow three argument representations for prop but only if the third argument is an event index: prop(x, $x1$, I): there is an event I which states that prop(x, $x1$) (cf. Sections 3.2 and 6.3). In tools such as AVERAGE, COMP and COMP_PROP (see below), for the sake of convenience, prop is represented without its arguments x and $x1$, since the position of x and $x1$ is not ambiguous in it. However, when prop has three arguments, it must be represented with its arguments because either x or I may be referred to (cf. Section 6.3).

A further extension is to allow a more complex logical formula instead of simply the notation prop. This point is introduced in Section 6.3 and will be developed in forthcoming work.

The next predicate we introduce is COMP(prop, $s1$, $s2$, $x1$, $x2$, $F(x1, m(x2))$), which is used to compare two sets of objects for a given property prop, and where:

- $s1$, $s2$ are sets of objects (each set is represented by a list; they can be elementary sets, with only one element, or empty sets in particular cases),
- prop is a measurable property,
- $F(x1, m(x2))$ is a function where $x1$ and $x2$ represent respectively the values, for the property prop, of any element of $s1$ and any element of $s2$, m is an arithmetical function, and F is a PROLOG predicate: equal, ge (greater or equal), le (less or equal), etc.

The last argument of COMP can also be a conjuction of $F_i(x1, m_i(x2))$. More

formally, COMP is true iff

$$(\forall x \in s1)(\forall y \in s2)(\exists x1, x2 \in \mathbb{R})\text{prop}(x, x1) \wedge \text{prop}(y, x2) \Rightarrow F(x1, m(x2)).$$

Notice that we use the notation $F(\ ,\ )$ for convenience. F has to be transformed into PROLOG code before evaluating the formula.

The last predicate is COMP_PROP(prop1, prop2, s, $x1$, $x2$, $F(x1, m(x2))$), where:

- prop1 and prop2 are two measurable properties, expressed in the same units (e.g meters for height and length),
- s is a nonempty set of objects,
- $F(x1, m(x2))$ has the same meaning as above.

COMP_PROP is used to compare the values $x1$ and $x2$ of all the objects x of s for properties prop1 and prop2. COMP_PROP is true iff

$$(\forall x \in s)(\exists x1, x2 \in \mathbb{R})\text{prop1}(x, x1) \wedge \text{prop2}(x, x2) \Rightarrow F(x1, m(x2)).$$

3. REPRESENTATION OF NOUNS, VERBS, RELATIVE CLAUSES, AND INTERROGATIVES

Nouns, verbs, and relative clauses are represented in a way quite similar to [5], [6], and [31]. However, we give a more precise and explicit representation of nouns, noun modifiers, verb modifiers, and negation. Interrogatives are treated as in [31]. We also introduce the representation of ambiguities.

3.1. Representation of Noun Modifiers

There are two classes of noun modifiers: those that can be paraphrased by a relative clause and those that cannot [31, 21].

In the first case, the noun modifier is represented by a subtree that contains a verbal predicate. The subtree represents the relative clause the noun modifier is equivalent to. One of the advantages of such a representation is that the underlying semantic relationships between the head noun and the modifying noun are made more explicit.

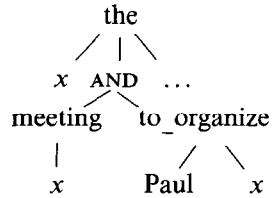
In the latter case, the noun modifier is represented by a one-place predicate, and the head noun by a two-place predicate. Two arguments are necessary in order to make it possible to establish a link between the head noun and its modifier. Examples:

"Paul's meeting..."

is equivalent, for instance, to

"the meeting that Paul organizes...".

It is represented by



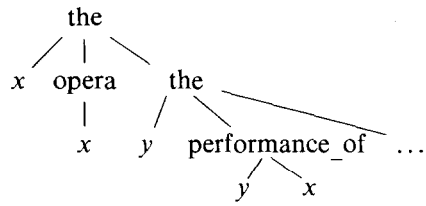
Consider now

"The performance of the opera...".

This sentence cannot be paraphrased by:

"The performance which is of the opera...".

It is represented by



Notice that, in this case, the variable x linked to "opera" has a wider scope over y.

The distinction between the two types of noun complements is done at the level of the dictionary. To each noun that may potentially be modified is associated a set of pairs composed of:

the semantic feature(s) of possible complements,

the corresponding semantic representations.

If several representations are possible, then contextual constraints may be used to select the correct one. For example, "Paul's meeting" may mean: (1) "the meeting organized by Paul" or (2) "the meeting about Paul". Constraints of selection make reference to facts, rules, and integrity constraints of the knowledge base. A representation is used if its constraints of selection are true or if it is a representation by default.

Nominal compounds behave in a way quite similar to noun complements, but with much more complexity. For example, consider "a woman doctor", "a Paris flight". The last example may mean a flight bound for Paris, a flight coming from Paris, or even a flight with a stop at Paris. Here, there is a complete lack of syntactic information to guide the semantic process. It is possible to define some rules which can capture quite general forms of modification between the modifying and the modified concept [13], but nominal compounds most of the time remain ambiguous and require the help of context.

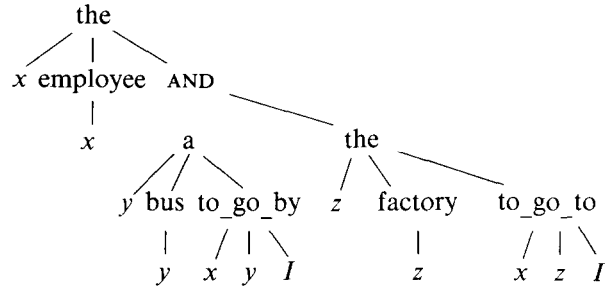
3.2. Representation of Verbs and Verb Complements

Verbs are represented in different ways according to their complements. Intransitive verbs have only one argument, which represents the subject of the verb. Transitive verbs may have two or three arguments. The representation with three arguments is used for verbs that have both a direct and an indirect object complement. In the other cases, verbs have two arguments. We do not treat completive propositions here, because they are not yet very common in natural-language front ends. In order to have a uniform representation and to be able to deal correctly with negation, each adverbial complement is represented separately. Thus, an index is added as a third (or fourth) argument to each verbal predicate so as to capture the fact that these predicates refer to a single event. The index (represented by an integer) is a

way of linking several predicates that describe a given event [23,25,19]. For instance,

"The employees go by bus to the factory."

is represented by



The event is represented here by the constant *I*.

More complex representations for events can be used to represent event sequencing depending on time, including the use of possible worlds for events in the future. Events can also be used for nouns when a head noun is modified by several noun complements.

Verbs can accept sets as arguments (see examples in Section 4). A set is represented by the list of its elements: *x*1.*x*2.....*xn*.nil. In this case, the evaluation on the knowledge base can hold only if sets are not ordered, i.e. *a.b.c*.nil is equivalent to *b.c.a*.nil, etc.

Ambiguities can appear at the level of prepositional phrase (PP) attachment. The problem is to know where to attach a PP: to the NP it follows (it is then a noun complement) or to the verb. This problem is illustrated by sentences such as

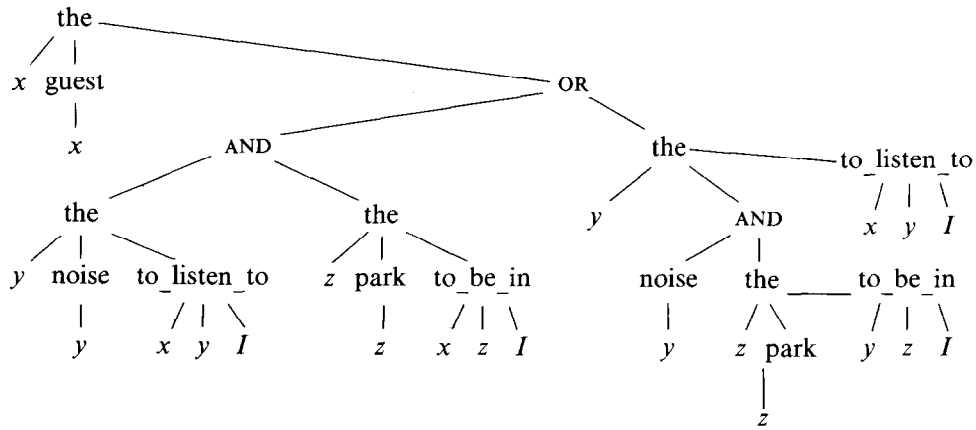
"Les invités entendaient le bruit de la fenêtre."

(The guests listen to the noise of/from the window.)

or

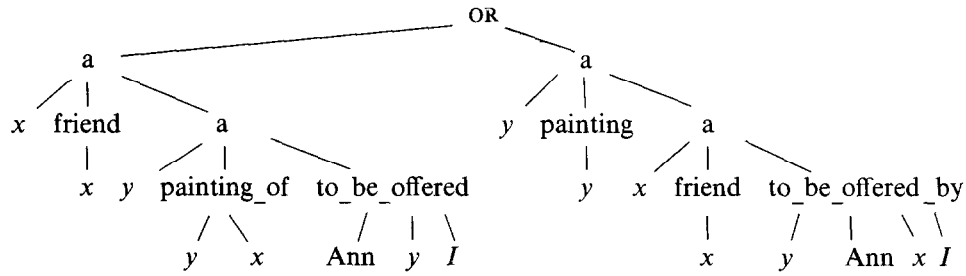
"The guests listen to the noise in the park."

In this latter example, the noise or the guests may be in the park. In a way quite similar to [17], we make explicit the ambiguities by representing each way of understanding a sentence (or a part of it) and by linking them with an OR. The latter example is represented by



Only one of the representations in this disjuction is correct. It is sometimes possible to eliminate some OR-branches at this level via appropriate references to the context. This is, in particular, the case when a branch violates an integrity constraint. However, this way of representing ambiguities will be used each time contextual information is not informative enough to allow unambiguous selection of the correct representation. Notice that the event index is the same in each branch since only one branch is correct.

"Ann was offered a painting by a friend." is also ambiguous; it depends on whether Ann's friend is the artist or the donor:



Notice that the subtree composed of a verb and of some of its complements may be rewritten into a more precise form much closer to the application domain, as in [30]. The two alternatives of the event stated in this example are represented using the same event index, since only one alternative is true.

The last point to examine is negation in verb phrases. For instance, the sentence

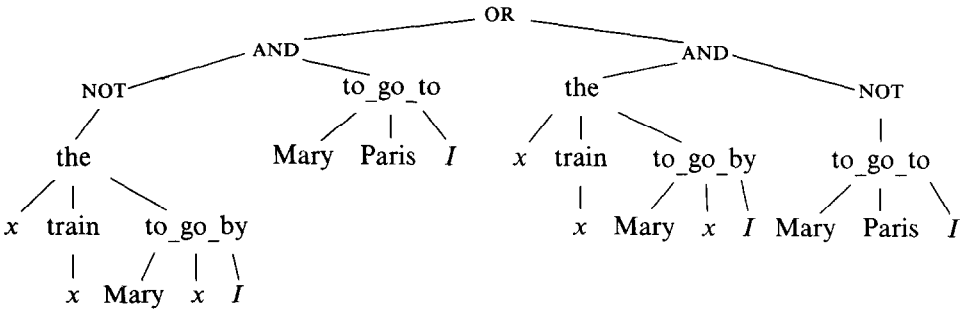
"Mary does not go by train to Paris."

may mean

(1) that Mary uses another means of travel than the train to go to Paris,

(2) that Mary goes by train but not to Paris.

Here again, we represent the different readings of the VP and link all the representations by an OR. The above sentence is represented by



From the focalizer point of view [25], we can say that the correct representation depends on which structure the negation is focussed. From that point of view also, there is a third reading of the sentence where the focus is on Mary. (This idea was suggested to us by M. McCord.)

At the level of the evaluation process, the method usually chosen to deal with negation is to augment the PROLOG SLD-resolution proof procedure with the negation-as-failure rule [3, 22, 35].

3.3. Representation of Relative Clauses

In traditional approaches, there are two kinds of relative clauses: those that introduce a restriction on the set of objects denoted by the noun to which they apply, and those that do not. This latter case is more difficult to characterize. However, it seems that most of them can be considered as a digression that explains some properties of the noun to which they apply. We call this latter class explicative relative clauses. Examples:

"J'ai rencontré des amis qui se promenaient."

(*I met friends that were walking.*)

and

"John who is fond of chamber music is going to a concert".

are both explicative relative clauses.

More formally, we say that a relative clause RC introduces a restriction on a variable x, at a given moment and in a given context, for a noun N and a determiner Det if and only if the set of objects for which the logical formula associated to the noun phrase is true is strictly included in the set of objects for which the logical formula associated to the noun phrase without this relative clause is true.

The logical form associated to a restrictive relative clause is an open formula with a variable under the scope of a quantifier introduced at the level of the head clause.

The semantic representation of a relative clause that introduces a restriction is included in the second branch under the appropriate node of the three-branched structure and linked to the noun it modifies by an AND, as in [5], [6], [7]. On the other hand, an explicative relative clause has to be processed as part of the clause immediately dominating it because it might introduce new information about the entity to which it is linked.

4. REPRESENTATION OF DETERMINERS

The role of determiners is to express the degree of determination of the noun phrase they precede. Thus, determiners introduce a quantification on the variable that represents the entity denoted by the noun they precede. A resulting problem is to represent the relations between the variables that appear in the semantic representation of a sentence. We have studied the problem of the range and the scope of quantified variables in [34]. We have defined a set of rewrite rules whose goal is to produce all the relevant quantifier configurations. Some rules are blocked if they contradict a fact, a rule, or an integrity constraint of the knowledge base.

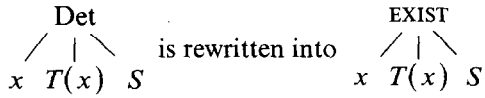
In this section, we examine how determiners are rewritten into a more precise semantic representation. We distinguish distributive and collective readings of a set

of entities. Most determiners have different representations depending on context and also on the entities that appear in the sentence. The rules we give below induce a distributive reading of a set of entities, in a way quite similar to [4], [6], [5], and [25]. In [5], [6], both distributive and collective readings are contemplated in the same formalism.

4.1. A distributive representation of determiners:

We now describe how determiners are represented. For each determiner Det or class of determiner, we show which new subtree the subtree whose root is this determiner is rewritten into. Our purpose is not to give an exhaustive list of possible representations of determiners, but rather to induce a possible method to write them. Notice also that scoping problems are solved before applying the transformations described below [34]. This point, added to the fact that contextual information may be used at the level of each possible representation of a determiner as well as at the level of a whole NP to select the correct representation of a determiner, allows us to associate to each determiner a set of translations *a priori* independently of the translation of the other determiners of the noun phrase, and by extension, of the whole sentence.

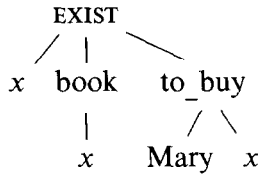
Rule D1. Det = {a, a few, several, some, ... }:



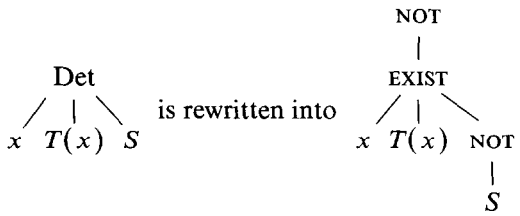
Example.

"Mary is buying a book."

is represented by



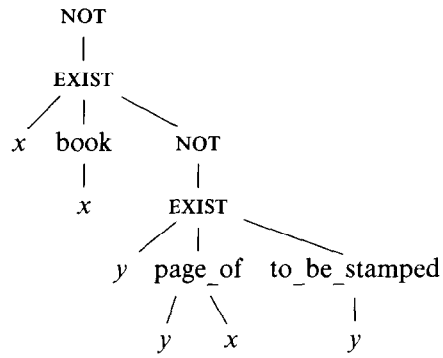
Rule D2. Det = {all, all the, each, every, the (plural), ... }:



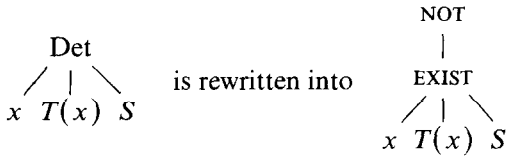
Example.

"A page of each book is stamped."

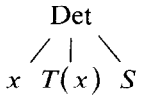
is represented by



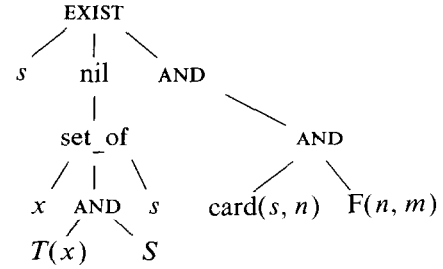
Rule D3. Det = {no, not, any, ...}:



Rule D4. Det = {one, two, three, at least four, less than three, a single, ...}:



is rewritten into



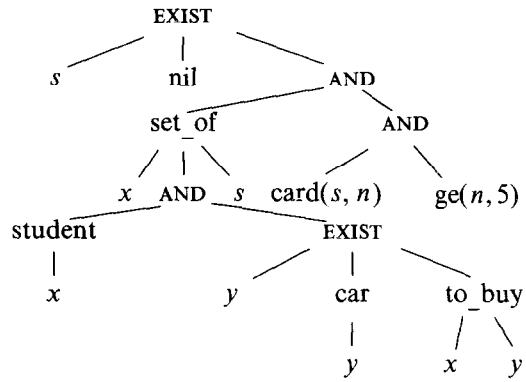
where

- m is a constant,
- F is a conjunction of PROLOG calls: ge, le, eq, diff, as described in Section 2.

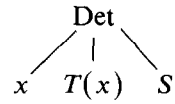
Example.

"At least five students have bought a car."

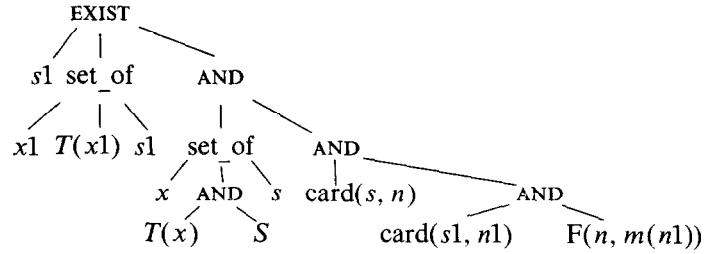
is represented by



Rule D5. Det = {many, almost, several, a few, a lot of, a majority, ... }



is rewritten into



where

- m is an arithmetical function, as defined in Section 2,
- $s1$ represents the set of entities denoted by the noun that follows Det, whereas
- s represents the set of objects which satisfy the whole semantic representation which is in the scope of Det.

We only give here one possible representation for determiners such as "many". However, it appears that, for instance, "many" has at least seven different representations. They will be presented in a forthcoming work.

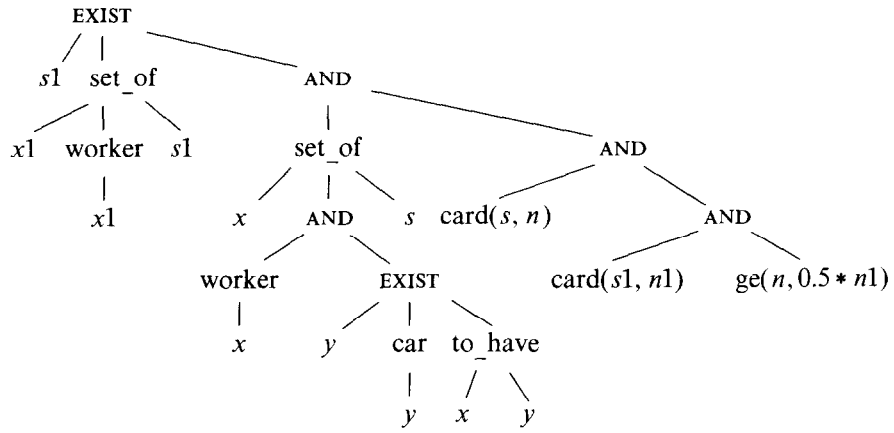
In [25], a point is made about the importance of quantification over several variables simultaneous, in a way which seems not to be as explicit as ours, but with an original and insightful emphasis on focus.

Notice that in rules D4 and D5, the second argument loses its restrictive role. The reason is not a theoretical one but only a consideration of efficiency. This presentation avoids reevaluating the same predicate.

Example.

"Many workers have a car."

is represented by

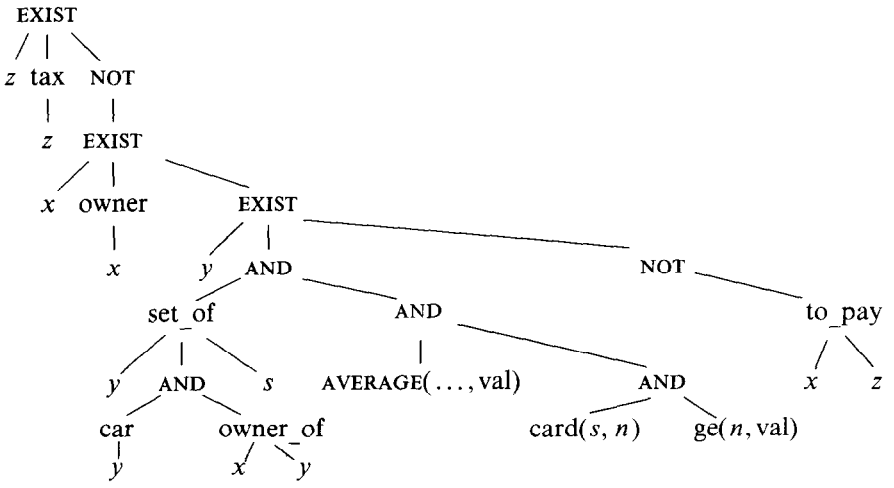


We represent "many" by stating that the number of workers that have a car is greater than half the number of workers known to the knowledge base.

The function F can also require the use of the predicate AVERAGE. For instance, to represent the following sentence (which we suppose not to be ambiguous):

"All the owners of many cars pay a tax."

it is necessary to compute the average number of cars per owner and then to select those who have more cars than this average number. The sentence is represented as follows:



where AVERAGE(..., val) is equal to

$$\text{AVERAGE}\Big(\text{card}, s1, \text{set\_of}\Big(s1, \text{AND}\Big(\text{owner}(x1), \text{set\_of}\Big(y1, \text{AND}\Big(\text{car}(y1), \text{owner\_of}(x1, y1)\Big), s1\Big)\Big), s2\Big), \text{val}\Big)$$

$s2$ is the set of sets $s1$ of cars of a given owner $x1$. AVERAGE computes the average cardinal of the sets $s1$.

The latter example above shows that several representations are possible for determiners such as several and many. In fact, the two main trends are to attach to those determiners an absolute or a relative meaning. By absolute meaning, we mean that the set of entities considered is viewed as a whole, as in "Most of the employees have a car": we consider "most of the" employees of the knowledge base. By relative meaning we mean that the set of entities we consider is structured in the form of a set of sets. Relative meaning requires the use of the average value of a set of objects for a given property. In both cases, the values assigned to the function F for a given determiner remain invariable. Our approach to relative meaning takes into account the specificities of the knowledge base with a real flexibility, because no value for a given determiner is assigned *a priori* to F.

In a recent work, M. McCord [27] came upon the distinction between relative and absolute meanings of such quantifiers. He found a way of representing both with a single general PROLOG predicate, the distinction between the two types of readings being made at the level of the specification of what is in the "base" and what is in the "focus" for a given sentence. However, it is sometimes difficult for a system to make a clear distinction between base and focus, so here only make explicit the different types of possible readings by distinct representations (including possible additional ambiguities). The difficulty of making a distinction between base and focus arises, for instance, in case of lack of intonation indication or in case of ambiguity about focus when several levels of focus are involved (e.g. main focus and subfocus).

Another point is that it is possible, depending on the domain, to assign to m in $F(n, m(n1))$ a different value for each determiner. Thus, a difference of degree is made between "several", "a majority", "many", etc. It is also possible to encode, by a conjunction and/or a disjunction of functions F, results of fuzzy set theory [39].

4.2. Using Sets as Arguments

We now consider sentences where a collective reading is necessary. Previous studies about this point that we have used as a basis for this work are [4], [5] and [7]. For instance, in sentences like

"Les droites parallèles ne se coupent pas."

(Parellel lines do not cross each other.)

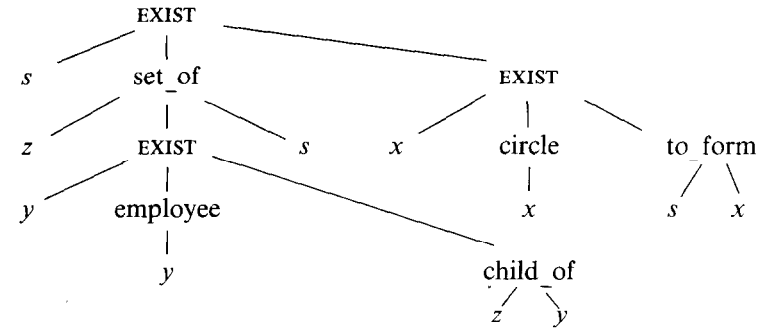
and

"Les enfants des employés forment un cercle."

(The children of employees form a circle.)

the subject argument of the verb is a set of objets. Thus, we extend the formalism described in Section 4.1 by allowing a variable to represent a set. The second

example above is represented as follows:



The French determiner "des" is viewed here in the sense of "some"; thus, it is represented by an existential quantification.

Notice that what we call a collective reading must not be confused with the sentences such as

"Peter and John have crossed the road."

where "the road" denotes a single entity that both Peter and John have crossed separately.

Finally, notice that a representation using sets introduces additional ambiguities in the definition of the set itself. In a sentence like

"Les enfants de chaque employé forment un cercle."

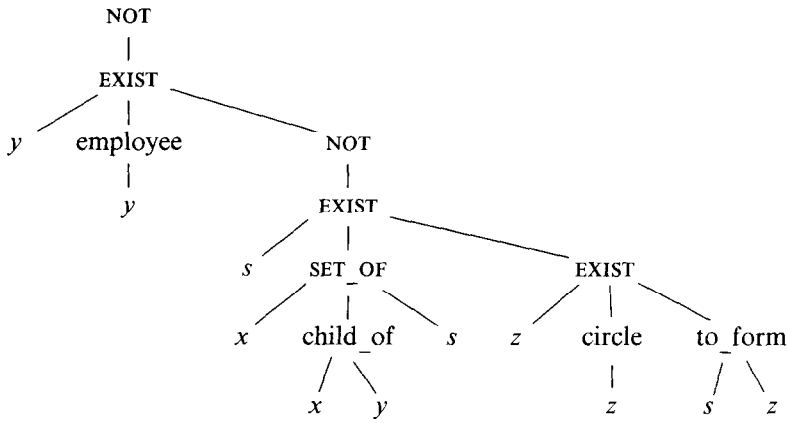
(The children of each employee form a circle.)

a circle may be formed by:

(1) the children of all the employees,

(2) the children of each employee, i.e., there are as many circles as there are employees.

The first alternative is represented above; the second is represented as follows:



5. REPRESENTATION OF ADJECTIVE PHRASES

We now examine the semantics of adjective phrases. We first give some definitions and then develop some examples.

5.1. *On Classifying Adjectives*

In a way quite similar to [21] and [37], we make a distinction between adjectives that introduce a restriction on the noun they apply to and those that do not. The class of restrictive adjectives is composed of two main subclasses:

Scalar adjectives, such as large, tall, good, bad, rapid, which are related to measurable properties related to nouns. For instance, tall may be directly linked to the property "height": height(John, 1.80).

Intersecting adjectives, such as French, circular, which determine a property that is *a priori* not inherent in the noun being qualified, but rather a property that is added to the noun. These adjectives allow one to characterize a subset of objects. They are represented by a predicate with one argument: European(Edith), and are related to Boolean properties.

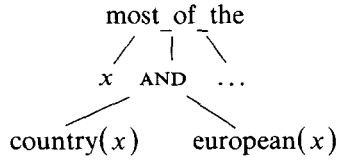
The nonrestrictive adjectives may also be subcategorized. We will only point out here the class of *negative adjectives* (fake, false, ungrammatical, artificial, etc.). They are nonrestrictive, since the object they apply to may not exist in the knowledge base. For instance, the concept of "artificial intelligence" does not entail that the concept of intelligence must be known to the knowledge base.

Depending on context, an adjective may belong to several classes. The distinction between these different classes of adjectives is fundamental, since, as we will see in the next sections, adjectives have very different representations depending on the subcategory they belong to. For example, "artificial" may be a scalar adjective if we are in the domain of psychology, where "a very artificial behavior" is an acceptable description. It may also be an intersecting adjective, as in "artificial flowers", or a negative adjective, as in "artificial intelligence". The representations of the two first cases are presented in the next sections. In the last case, the noun and the adjective are represented as a whole.

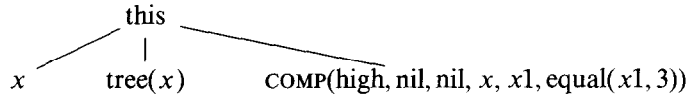
Finally, it is important to note that, depending on the noun it applies to, an adjective of a given class may also have different translations, as, for instance, "rapid" in "a rapid train" and "a rapid construction". Semantic types and contextual knowledge can be used to characterize the different representations of an adjective.

5.2. *Semantic Representation of Adjective Phrases*

In our semantic representation, intersecting adjectives are treated as conjoined predicates [5, 6, 7, 4]. Thus "Most of the European countries" is represented by

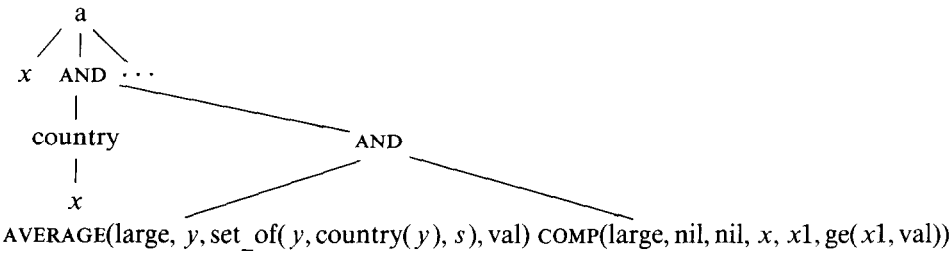


(For clarity, determiners are not rewritten into the representation.) The sentence "This tree is three meters high." is represented by

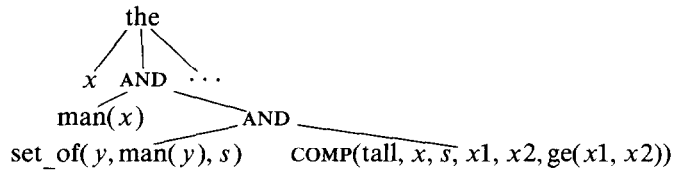


[$x1$ is the height of x: high(x, $x1$)].

Scalar adjectives are, in fact, implicit comparatives [2]. The surface NP "Det Adj(+scalar) N" means that the value for the property(ies) referred to by the adjective for each object that satisfy the logical formula associated to "Det Adj N" is higher than the corresponding average value for all the objects that satisfy the logical formula associated to N. Thus, a NP like "a large country" is represented by



Superlatives, as in "the tallest man", are represented as follows:



In this representation, we express that x is taller than any man y. It is essential to note that in tall(x, y) we exhibit some measure y of tallness, but that does not entail that x is tall. Having tallness and being tall are distinct concepts.

Notice that superlatives do not apply just to the noun they precede, but rather to the part of the NP which is in the scope of the determiner that precede the superlative [31]. In

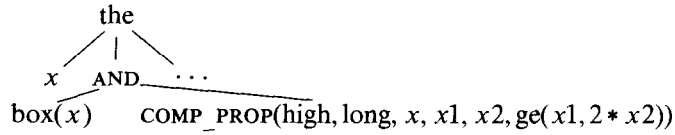
"The most famous musician that plays the violoncello"

"most famous" refers to the set of musicians that play the violoncello rather than to the most famous of all the musicians. We assume here that the relative clause is restrictive, which seems highly probable.

Comparisons between two properties can also be described by our formalism:

"the boxes two times higher than long..."

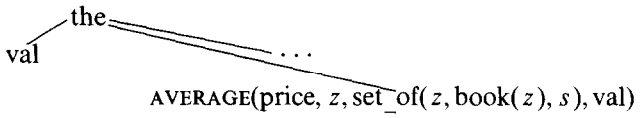
is represented by



Notice that the last argument of COMP may be a conjunction of predicates: "Paul is between 2 and 3 times richer than John." is represented by

$$\text{COMP}(\text{rich}, \text{Paul}, \text{John}, x1, x2, \text{AND}(\text{ge}(x1, 2 * x2), \text{le}(x1, 3 * x2)))$$

The nonrestrictive adjective "average" can also be represented by our formalism. The NP "the average price of a book" is represented by



where "price" is considered as a property of a book. In all the constructions of the form "The average X of Y", X is, in fact, a property of Y, expressed by a noun.

In some contexts, adjectives such as "nervous", "presidential", or "parallel" applied to some nouns are equivalent to noun complements or to more complex domain dependent structures:

"the presidential election" is equivalent to "the election of the president";

"the nervous system" is equivalent to "the system of the nerves".

A possible representation for "parallel" is a predicate with a set (or sets) as arguments(s).

Transitive adjective phrases, such as fond of, afraid of, fed up with, are represented as being the main verb of a clause when they are attributes and by a relative clause in the other cases.

Finally, adjectives such as "perfect", "worst", "ideal", "average" (in "the average X") introduce various kinds of comparisons on a set of properties. In fact, each of these adjectives has a representation that depends on the noun it applies to. These representations are expressed in terms of a conjunction (or disjunction) of COMP (or COMP_PROP) for some properties expressed by scalar adjectives and the existence or the absence of other properties expressed by intersective adjectives.

Notice that the scope of adjectives can sometimes be quite complex to determine, as in

"Three two horse carriages have arrived."

which should be parenthesized:

"(Three (two horse) carriages) have arrived."

(See other examples in [38].)

The semantic representation of an adjective phrase is the composition of the partially instantiated representations of the elements it is composed of (in particular, if the adjective is in the superlative or comparative form or if it is modified by an adverb). Additional instantiations will be carried out when the representation of the adjective phrase is included in that of the noun phrase it is an element of.

6. REPRESENTATION OF ADVERBS

Contrary to a common idea, adverbs are very often used in a real man-machine communication. In particular, adverbs that express an idea of quality (quickly, well,

early, etc.) or an idea of intensity or of degree (very, enough, about, at least, much more, etc.), and those that introduce temporal (e.g. yesterday, very soon, etc.) and locative references, have to be taken into account in any natural-language front end. In this work, we will concentrate on the two first types of adverbs.

In [25], M. McCord proposes some ideas about a possible representation for temporal adverbs. He also studies a class of adverbs called quantificational adverbs (such as: often, usually, seldom, never), which he paraphrases using quantificational determiners. For instance, "John always buys books at Smith's" with focus on "books" is equivalent to "All John's purchases at Smith's are books." This approach is undoubtedly very promising; however, the problem of transforming the first sentence into the latter is not very clear at present and will probably involve quite complex theoretical problems.

In order to go forward and to simplify the problem, we will consider that:

Adverbs of intensity apply to adjectives and to adverbs of quality. They are used to refine, reinforce, or weaken the meaning of the structure to which they apply.

Adverbs of quality apply to verbs. The use of an adverb of quality is a way to restrict (or to impose constraints on) the meaning of a verb according to one or more of its characteristics without using explicit complements. For instance, we can say

"John gets up early."

instead of

"John gets up at 6 A.M."

From that point of view, we can say that adverbs operate on verbs in a way similar to adjectives on nouns.

It is commonly agreed that most adverbs introduce higher-order predicates that apply to one or more first-order predicates. This approach seems fruitless in the sense that it only postpones problems if a more precise semantics is not associated to adverbial predicates. In our work, we simply show how some of the most useful adverbs of quality and intensity in some precise context can be represented by the tools developed above, i.e. with a first-order specification.

Before going forward, we take note of a specific class of adverbs that apply to events or actions, as for instance in

"It is the first time that $P(\ldots, I)$,"

where I is an event index. We would roughly represent this sentence by stating that there are no events J, strictly smaller than I (events are numbered in order), such that $P(\ldots, J)$ is true.

In our natural-language understanding system, the meaning of an adverb, may (1) differ from one subset of a language to another subset and (2) depend on the semantic features of the NP or the VP in which it occurs.

In the next subsections, we first show how adverbs applied to adjectives can be represented. Then, we examine the representation of adverbs that apply to verbs. We make a distinction between those that are directly related to an adverbial property of the verb (i.e. which are more or less equivalent to an adverbial complement) and

those that express the frequency of an event (e.g. "often"). Finally, we show how some scoping problems can be taken into account. This latter point will lead us to introduce some extensions to the original tools presented above.

6.1. Adverbs in Adjective Phrases

The semantics of adverbs of intensity applied to adjective phrases can be expressed in two ways:

(1) By a modification of the function $F(\ )$ [39]. The kind of modification they involve depends on the adverb. We distinguish three types of modification:

Adverbs, such as "very", that modify the value of m in $F(x1, m(x2))$. For example, "a large house" can be represented by stating that the number $(x1)$ of its rooms is higher than the average number $(x2)$ of rooms of a house: $\text{ge}(x1, x2)$. Then, "a very large house" can be represented by stating that the number of its rooms is, for instance, two times higher than the average number of rooms of a large house.

Adverbs, such as "at least", that modify F. For instance, "Paul is three times richer than Mary" is represented by

$$\text{COMP}(\text{rich}, \text{Paul}, \text{Mary}, x1, x2, \text{eq}(x1, 3 * x2)),$$

whereas "Paul is at least 3 times richer than Mary" is represented by

$$\text{COMP}(\text{rich}, \text{Paul}, \text{Mary}, x1, x2, \text{ge}(x1, 3 * x2)).$$

Adverbs such as "rather" or "quite" that imply that complementary predicates are added to F (which itself remains unchanged). For instance, "tall men" denotes the set of men whose height is greater than the value of the average of the height of men; the adjunct of "rather" restricts this set by adding, for example, the constraint that these men are smaller than two times the value of the average value.

(2) By using a conjunction of several subtrees AND(AVERAGE, COMP) or COMP. To be more precise, let prop be the property involved in the adjective representation; then, a first value val1 for prop is computed by AVERAGE on a set $S1$. COMP entails the definition of $S2$, $S2$ being strictly included in $S1$. Next, a second value val2 for prop is computed by AVERAGE on the set $S2$. COMP then builds $S3$ from $S2$. This process may be repeated any number of times.

Adverbs such as "rather" or "very" can be represented in this way. For instance, "rather tall men" can be represented by stating that the height of these men is greater than the average size of men but less than the average height of the tall men. This second alternative is less efficient (from the evaluation point of view) than the first one, but it takes into account the specificities of the knowledge base with a greater flexibility.

6.2. Adverbs in Verb Phrases

Within our framework, only adverbs of quality that can be directly related to complements that describe a measurable property of a verb can be taken into

account. Thus, for example, if a sentence like

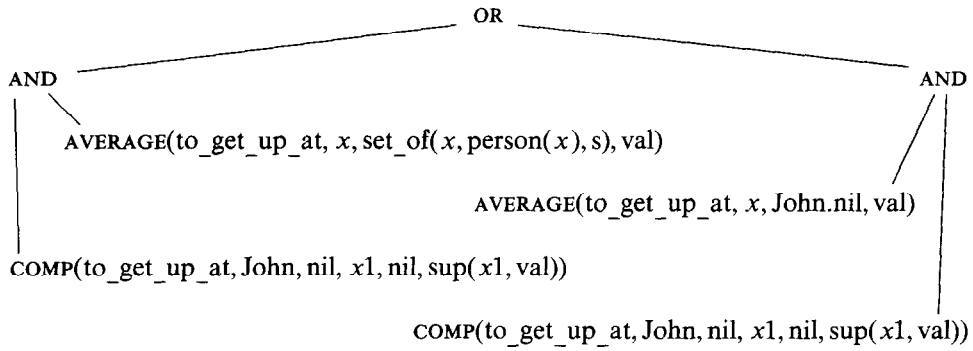
"John gets up at 6 A.M."

is represented by

to_get_up_at(John, 6, I)

(the index I represents the event of John's rising), then the sentence "John gets up early." can be treated, because the adverb "early" is related to a complement (the hour) which is measurable. In this case, the semantic representation of an adverb is very similar to that of an adjective.

To represent "early", it is necessary to compute the average value of the rising hours associated to the property "to_get_up_at" for all the humans for which the rising hour is known to the knowledge base. Then, we have to compare it with John's rising hour. There is also another way of understanding this sentence. It may mean that John gets up earlier than he is used to. In this case, the average value is computed for the property "to_get_up_at" from the set of all John's rising hours known to the knowledge base. Thus the final representation of this ambiguous sentence is:



Notice that, according to the semantics of AVERAGE and COMP described in Section 2, when AVERAGE(to_get_up_at, x, John.nil, val) is computed, all the facts to_get_up_at(John, _, Ei) are taken into account, whereas the value $x1$ taken into account in COMP is the value of to_get_up_at(John, x, I) where I is the event referred to when we say that "John gets up early". The last event can be taken into account automatically by the evaluator. The event notation can also be added to COMP, in a specific argument. This latter solution is more general, since it allows one to deal with events in the past and does not entail an arbitrary ordering of facts in the knowledge base.

The other class of adverbs we consider here are those which are related to the frequency of an action (e.g. often, rarely). To take such adverbs into account, it is necessary to be able to count, in the knowledge base, the number of occurrences of an action, possibly under some constraints (expressed by complements). For instance, consider the following independent statements:

"Mary comes."

"John comes by train."

"John comes with Sue."

which are represented by

to_come(Mary, 1).

EXIST$(x, \text{train}(x), \text{to_come_by}(\text{John}, x, 2))$

to_come_with(John, Sue, 3).

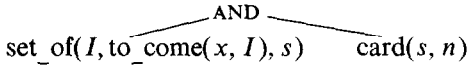
These three events (denoted 1, 2, and 3) are stored in the knowledge base. From these statements it is possible to deduce that:

4 people have realized the action of coming,

John has come two times,

John has come only once by train.

In this context, the sentence "John come often." can be represented by stating that the number of events "John comes" is higher than the average number of comings for all the other persons that have done the action of coming. The number of actions of coming for any person x is computed by

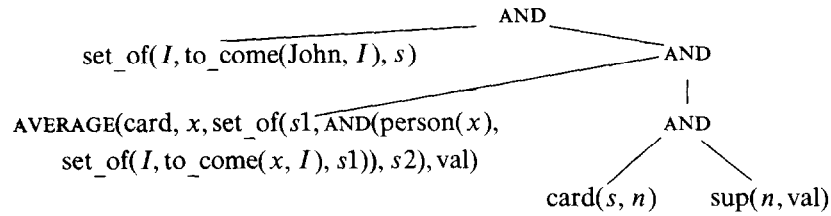


(I is a variable that denotes events).

In our example, if x = John, then s = 2.3.nil. As we need to include repetitions, the PROLOG predicate list_of is used rather that set_of. The average number of actions of coming is obtained by

AVERAGE$(\text{card}, x, \text{set_of}(s1, \text{AND}(\text{person}(x), \text{set_of}(I, \text{to_come}(x, I), s1)), s2), \text{val})$.

where $s2$ is a set of sets $s1$. The final representation of the sentence is



The representation of a verb phrase with complements is based on the same idea. "John comes often to Paris." means that John comes more often to Paris than the average comings to Paris for all the other persons who come to Paris. Notice that, in our previous examples, we have not introduced any notion of time in the knowledge base, but only that of event.

6.3. Scoping Ambiguities

The next point to examine is the scoping ambiguities introduced by adverbs in a VP with NP or PP complements. This difficulty has already been pointed out in [2] and

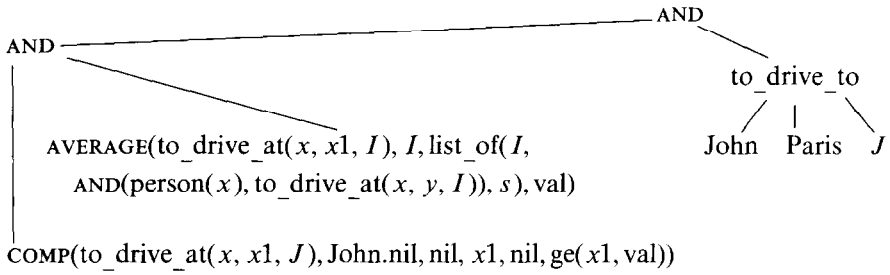
in [25]. Consider, for example, the two sentences

"John is driving quickly to Paris."

"John is driving quickly on the snow."

In the first sentence the scope of "quickly" is "is driving", whereas in the second sentence it is the whole VP "is driving quickly on the snow". In addition, notice that "John is driving quickly" is ambiguous (see Section 6.2). For simplicity, in this subsection we will not take this ambiguity into account and compare John's driving with the driving of other persons.

In the first example, the two modifiers of the verb, "quickly" and "to Paris", are represented separately, as two adverbial complements



where the predicate "to_drive_at" has three arguments, the first representing the driver, the second the speed, and the third the event

to_drive_at(John, 140, J).

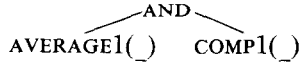
The second sentence above means that John drives on the snow quicker than the average speed of driving on the snow of other persons. This sentence cannot be directly represented by COMP and AVERAGE, because the first argument of these predicates is a single predicate. What we need here is to have a full formula as the first argument of COMP and AVERAGE. Informally, this full formula contains (1) the predicate prop associated to the property about which an average value is computed and (2) additional predicates that restrict prop. The general form of this formula is a three-branched quantified tree. Thus, the average driving speed on the snow of a person is computed by AVERAGE1 as follows:

AVERAGE1(the(y, snow(y), the(x, person(x), AND(to_drive_on(x, y, i),
to_drive_at(x, z, i)))), z, list_of($x1$, the($y1$, snow($y1$), the($x1$, person($x1$),
to_drive_on($x1$, $y1$, i))), s), val)

and the comparison with John's driving speed is done by using COMP1:

COMP1(the(y, snow(y), AND(to_drive_on(John, y, i),
to_drive_at(John, z, i))), John.nil, nil, sup(z, val))

and the final representation is



This approach of the representation of adverbs in VPs can be extended to VPs with several complements. However, with several complements, the number of ambiguities may increase drastically.

6.4. Adverbs of Intensity Applied to Adverbs of Quality

The last point of this section is to show how adverbs of intensity can modify adverbs of quality in VPs. In fact, adverbs of intensity applied to adverbs of quantity modify, and thus are represented, in the same way as when they apply to adjectives. Thus, they are represented by a modification of the function F of the original semantic representation or by the adjunction of subtrees AND(AVERAGE, COMP) (see Section 6.1).

7. DISCUSSION AND CONCLUSION

In this paper, we have presented the main lines of our approach to the semantics of natural-language sentences in logic programming. We have defined tools, based on first-order logic augmented by PROLOG calls, which an expert can use to describe the semantics of each of the lexical entities involved in the domain he considers. The degree of precision of these tools should be appropriate for simple man-machine communication in natural language. The tools developed do not involve very complex computations, and thus the evaluation of the semantic representation on a knowledge base defined on a limited domain may be reasonably efficient. Several insightful ideas and tools about this point are given in [36].

The main characteristics of the formalism of the tools developed and illustrated here are (1) the closeness of its semantics to that of PROLOG and the use of a three-branched quantified tree, and (2) its ability to represent ambiguitites, to use sets as arguments, and to deal with scoping problems. However, this work should rather be considered as introducing a methodology that an exhaustive formalism. For that purpose, we have borrowed from set theory some basic and practical concepts, and we have described a way of representing noun and verb complements, negation, determiners, adjective phrases, and some adverbs. This work is being implemented for French, and it will be connected very soon to a syntactic parser that uses the formalism of gapping grammars [8, 9]. The implementation in itself is not problematic, but it raises several methodological problems: general tools to compute this representation, like those outlines in [12], in [27], and in [1], are strongly needed in order to have modular and clear grammar rules.

We have also raised problems about the way to compute the semantic representation of a sentence. Our central idea is to use rewrite systems where contextual conditions of application may be added to each rule. This point of view reflects the fact that the construction of the semantic representation of a sentence is not strictly compositional, in the sense attributed to Frege. To compute the representation of a sentence, it is necessary to take into account (1) the general context of its utterance, (2) the semantics, and (3) the position of words in it. These facts reveal the extreme complexity of (1) the semantic representation computation process and (2) the specification of a lexicon by a human expert. On this latter point, a general methodology and an adequate specification language have to be investigated in the future. An approach which seems very promising is to extract a sublexicon from the complete lexicon of a language and to add to each lexical item its corresponding semantic representation(s).

I would like to thank M. McCord, H. Abramson, V. Dahl, D. Coulon, G. Lapalme, L. Trilling, and J. Virbel for insightful and detailed comments about this work.

This work was supported by an ATP-CNRS and by the INRIA; both are public and civilian institutions.

REFERENCES

1. Abramson, H., Definite Clause Translation Grammars, in: *Proceedings of the International Symposium on Logic Programming*, Atlantic City, N.J., 1984.
2. Cercone, N., A Note on Representing Adjectives and Adverbs, presented at 5th IJCAI, 1977.
3. Clark, K. L., Negation as Failure, in: H. Gallaire and J. Minker (eds.), *Logic and Databases*, Plenum, New York, 1978.
4. Colmerauer, A., An Interesting Subset of Natural Language, in Clark and Tarnlund (eds.), *Logic Programming*, Academic, 1982.
5. Dahl, V., Un système déductif d'interrogation de banques de données en espagnol, Thèse, Univ. de Marseille-Luminy, GIA, 1977.
6. Dahl, V., Quantification in a three-valued logic for natural language question-answering systems, presented at 6th IJCAI, Tokyo, 1979.
7. Dahl, V., Translating Spanish into logic through logic, *Comput. Linguistics*, 7:3 (1981).
8. Dahl, V. and Abramson, H., On Gapping Grammars, in: *Proceedings of the 2nd Logic Programming Conference*, Uppsala Univ., 1984.
9. Dahl, V., More on Gapping Grammars, presented at Conference on Fifth Generation Computer Systems, Tokyo, 1984.
10. Dahl, V. and Saint-Dizier, P., Constrained Discontinuous Grammars, to appear.
11. Dowty, D. R., *Word Meaning and Montague Semantics*, Reidel, 1979.
12. Filgueiras, M. and Porto, A., Natural Language Semantics: A Logic Programming Approach, in *Proceedings of the* 1984 *International Symposium on Logic Programming*, Atlantic City, 1984.
13. Finin, T. W., The Interpretation of Nominal Compounds in Discourse, Technical Report MS CIS 1982-3, Univ. of Pennsylvania, Philadelphia, 1982.
14. Finin, T. W. and Palmer, M. S., Parsing with Logical Variables, presented at Conference on Applied Natural Language Processing, 1983.
15. Gallaire, H. and Minker, J., *Logic and Databases: A Deductive Approach*, Computing Survey, 1983.
16. Gross, M., Lexicon Grammars, in: *Proceedings of COLING*-84, 1984.
17. Hobbs, J. R., Representing Ambiguity, in: *Proceedings of the 1st West Coast Conference on Formal Linguistics*, 1982.
18. Hobbs, J. R., An Improper Treatment of Quantification in Ordinary English, in: *Proceedings of ACL*83, 1983.
19. Hobbs, J. R., Ontological Promiscuity, in: *Proceedings of the 23rd ACL*, Chicago, 1984.
20. Kamp, H. A Theory of Truth and Semantic Representation, in: J. Groendijk et al. (eds.): *Formal Methods for the Study of Language*, Amsterdam Press, 1983.
21. Keenan, E. L. and Faltz, L. M., *Boolean Semantics for Natural Language*, Vol. 23, Reidel, 1985.
22. Lloyd, J. W., *Foundations of Logic Programming*, Springer, 1984.
23. Moore, R. C., Problems in Logical Form, SRI Technical Note 241, 1981.
24. McCord, M., Using Slots and Modifiers in Logic Grammars for Natural Language, Technical Report 69-80, 1980; *Artificial Intelligence*, 1982.
25. McCord, M., Focalizers, the scoping problem and semantic interpretation rules in logic grammars, Tech. Report, Univ. of Kentucky, 1981; also in: D. Warren and M. van Caneghem (eds.), *Logic Programming and Its Applications*, Ablex, 1986.
26. McCord, M., Semantic Interpretation for the EPISTLE system, in: *Proceedings of the Second International Conference on Logic Programming*, Uppsala Univ., 1984.
27. McCord, M., Modular Logic Grammars, in: *Proceedings of 23rd ACL Meeting, Chicago*, 1985.
28. Mukai, K., Unification over complex indeterminates in PROLOG, ICOT Research Report No. 113, 1985; *F.G.C.S. J.* to appear.
29. Finin, T. W. and Palmer, M., Parsing with logical Variables, presented at Conference on Applied Natural Language Processing, 1983.

30. Palmer, M., Driving Semantics for a Limited Domain, Ph.D. Thesis, Univ. of Edinburgh, 1984.
31. Pereira, F., Logic for Natural Language Analysis, Technical Note No. 275, SRI International, 1983.
32. Sabatier, P., Dialogues en Français avec un Ordinateur, Thèse de troisième cycle, GIA Univ. d'Aix-Marseille, 1980.
33. Saint-Dizier, P., Modelling Human Computer Interactions in a Friendly Man-Machine Interface, in: *Proceedings of the Workshop on Logic Programming*, Albufeira, Portugal, 1983.
34. Saint-Dizier, P., Handling quantifier scoping ambiguities in a semantic representation of natural language sentences, in: V. Dahl and P. Saint-Dizier (eds.), *Natural Language Understanding and Logic Programming*, North Holland, 1985.
35. Van Emdem, M. H. and Kowalsky, R. A., The Semantics of Predicate Logic as a Programming Language, Comm. *ACM* 23:4 (1976).
36. Warren, D. H., Efficient processing of interactive relational database queries expressed in PROLOG, in: *Proceedings of VLDB*-81, 1981.
37. Woods, W., Kaplan, R., and Nash-Webber, B., The LUNAR Science Natural Language Information System, BBN Report 2378, 1972.
38. Yawar, A., Understanding Adjectives, Technical Report CSRI 167, Univ. of Toronto, Computer Systems Research Institute, 1985.
39. Zadeh, L. A., Test-Score Semantics for Natural Languages and Meaning-Representations via PRUF, Techical Note 247, AI Center, SRI International, 1981.